

EQEmu Perl Quest Lexicon

By: MWMDRAGON (mwmdragon@hotmail.com)

V - 03.05

Commands

quest::say

Explanation : Makes the mob or NPC say the text in the quotes.

Full Command : `quest::say("text");`

Text = What you want the mob or NPC to say.

Example:

Makes the mob greet the player, when hailed.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("Hello fellow Norrathian.");
    }
}
```

quest::emote

Explanation: Makes the mob or NPC emote the text in the quotes.

Full Command: quest::emote("text");

Text = What you want the mob or NPC to emote to the player.

Example:

Makes the mob emote the shaking of the head in disappointment, when hailed.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::emote("Shakes his head in disappointment.");
    }
}
```

quest::shout

Explanation: Makes the mob or NPC shout the text in the quotes.

Full Command: `quest::shout("text");`

Text = What you want the mob or NPC to shout out loud.

Example:

Shouts at the player to leave the city, when hailed.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::shout("Get out of our City!!!");
    }
}
```

quest::spawn

Explanation: Spawns a mob or NPC at the location given on the grid given.

Full Command: `quest::spawn(npc_type, grid, guildwarset,x,y,z);`

Npc_type = The mob or NPC's ID from your database.

Grid = The grid ID you want the mob or NPC to walk along from your database. Leave it as 0 if you don't want it to walk a grid

Guildwarset = set this as 0 unless you are running a guildwars server. It will assign it to a certain guild faction.

X,Y,Z = Are the exact spawn location for the mob or NPC.

Example:

```
# When hailed a new mob with the ID 7203 will spawn
# and walk along grid 24
# with no guild faction
# at the location -807,-124,-142

sub EVENT_ATTACK
{
    if($text =~ /hail/i)
    {
        quest::spawn("7203", "24", "0", "-807", "-124", "-142");
    }
}
```

quest::echo

Explanation: Sends the text in the quotes to the EQEmu console.

Full Command: quest::echo("text")

Text = What text you want the mob or NPC to send to the console.

Example:

Jokingly makes the NPC ask the console what he should do after being hailed

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::echo("Some player just hailed me...What do I do now?");
    }
}
```

quest::summonitem

Explanation: Summons an item for the player who triggers the event.

Full Command: quest::summonitem(itemid,amount or charges)

Itemid = The items ID number from your database.

Charges or Amount = Sets the amount of the item given to the player, or if the item can have a certain amount of charges set on it, it will set charges instead of amount of the item given.

Example:

Gives the player one instance of the item indicated after being hailed.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::summonitem("5120");
    }
}
```

quest::castspell

Explanation: Makes the mob or NPC cast a certain spell on the creature with the indicated ID.

Full Command: `quest::castspell(id,spellid)`

ID = The ID of the creature the spell is to be cast on.

SpellID = The ID of the spell to cast taken from the spells_us.txt

Example:

Casts the bind affinity spell on the hailing person.

```
sub EVENT_SAY
{
    if($text=~ /hail/i)
    {
        quest::castspell($userid,2049);
    }
}
```

quest::selfcast

Explanation: Makes the player cast a spell on themselves.

Full Command: quest::selfcast(spellid)

SpellID = The ID of the spell to cast taken from the spells_us.txt

Example:

After hailing, the player casts bind affinity on themselves.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::selfcast($userid,2049);
    }
}
```

quest::depop

Explanation: Makes the mob or NPC depop or unspawn.

Full Command: quest::depop()

Example:

After being hailed, the NPC depops.

```
sub EVENT_SAY
{
    if($text~/hail/i)
    {
        quest::depop();
    }
}
```

quest::cumflag

Explanation: Flag 50 for mob will increase by 1.

Full Command: quest::cumflag()

Example:

After being hailing, the NPCs Flag 50 goes up by 1

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::cumflag();
    }
}
```

quest::flagnpc

Explanation: Sets flag values on mobs and NPCs mainly for quests.

Full Command: quest::flagnpc(flag,flagvalue)

Flag = a variable you wish to assign the value to.

Flagvalue = the value you want the flag variable to equal

Example:

After being hailed, a flag on the NPC 7 is set to 1.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::flagnpc(7,1);
    }
}
```

quest::flagclient

Explanation: Sets a flag on the player, mainly for quests.

Full Command: quest::flagclient(flag,flagvalue)

Flag = a variable you wish to assign the value to.

Flagvalue = the value you want the flag variable to equal

Example:

After being hailed the NPC sets a flag on the player 3 to 1.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::flagclient(3,1);
    }
}
```

quest::exp

Explanation: Adds an amount of experience to the players character.

Full Command: `quest::exp(amount)`

Amount = The amount of experience you would like the player to gain.

Example:

After hailing the NPC the player gains 100 experience.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::exp(100);
    }
}
```

quest::level

Explanation: Sets the players character level to the level you specify.

Full Command: quest::level(newlevel)

Newlevel = The level you want the character to be.

Example:

After hailing the NPC the player is forced to level 25.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::level(25);
    }
}
```

quest::safemove

Explanation: Moves the player to the safe coordinates on the zone they are in.

Full Command: quest::safemove()

Example:

After hailing the NPC the player is move to the safe coordinates of the same zone.

```
sub EVENT_SAY
{
    if($text~/hail/i)
    {
        quest::safemove()
    }
}
```

quest::rain

Explanation: Makes it rain in the zone.

Full Command: quest::rain(*n*)

N = choose 1 to turn the rain on in the zone or 0 to turns the rain off in the zone.

Example:

After hailing the NPC it begins to rain in the zone.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::rain(1);
    }
}
```

quest::snow

Explanation: Makes it snow in the zone.

Full Command: quest::snow(n)

N= choose 1 to turn the snow on in the zone or 0 to turns the snow off in the zone.

Example:

After hailing the NPC it begins to snow in the zone.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::snow(1);
    }
}
```

quest::givecash

Explanation: Gives the player money based on whatever denominations you choose.

Full Command: quest::givecash(c,s,g,p)

c = the amount of copper to give the player.

s = the amount of silver to give the player.

g = the amount of gold to give the player.

p = the amount of platinum to give the player.

Example:

The NPC give the player 1 of each coin for hailing him.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        Quest::givecash("1","1","1","1");
    }
}
```

quest::pvp

Explanation: Turns Player Versus Player (PVP) on or off for the player.

Full Command: quest::pvp("onoff")

onoff = set this to ON for the player to be pvp enabled and OFF for the player to be non pvp.

Example:

After hailing the NPC the player becomes Player Versus Player enabled..

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::pvp("on");
    }
}
```

quest::doanim

Explanation: The mob or NPC will perform the animation you set it to do.

Full Command: quest::doanim(n)

n = See Appendix I (Animations) for possible animation numbers to set this to.

Example:

After being hailed the NPC begins to dance.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::doanim(58);
    }
}
```

quest::addskill

Explanation: Increases the players skill in the skill you choose by the amount you choose.

Full Command: `quest::addskill(skill,value)`

skill = See Appendix II (Skills) for a list of possible skill numbers to use.

value = the amount that you want to be added to the player skill.

Example:

After hailing the NPC the players baking skill is increased by 5.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::addskill(60,5);
    }
}
```

quest::me

Explanation: Makes the client do a nameless or ownerless emote as if from a narrator.

Full Command: quest::me("text")

text = the text in the quotes is what you want to use as an ownerless or nameless emote.

Example:

After hailing the NPC the player receives a emote as if from a narrator.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::me("The ground begins to shake beneath your feet");
    }
}
```

quest::faction

Explanation: Adjusts the players faction with a chosen faction.

Full Command: `quest::faction(factionid,value)`

Factionid = The id of the faction you would like to adjust for the player, taken from your database.

Value = The amount you wish to adjust the players faction with the chosen faction.

Example:

After hailing the NPC the players faction with faction ID 25 is increased by 5.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::faction(25,5);
    }
}
```

quest::setguild

Explanation: Adds a player to a certain guild and also sets their rank in that guild.

Full Command: `quest::setguild(guildid,rank)`

Guildid = The guild you want the player to be a member of, can be found in the database.

Rank = The rank you want the player to be. You can find or create this in your database.

Example:

After hailing the NPC the players is now the leader of the guild number 3

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::setguild(3,0);
    }
}
```

quest::rebind

Explanation: Binds the player to a chosen zone at a chosen location.

Full Command: `quest::rebind(zoneid,x,y,z)`

Zoneid = The ID of the zone you want to bind the player in. Can be found in your database.

X = The x Axis you would like to bind the player to. Can be found by /loc but switch the x and y.

Y = The y Axis you would like to bind the player to. Can be found by /loc but switch the x and y.

Z = The z Axis you would like to bind the player to. Can be found by /loc.

Example:

After hailing the NPC the players is bound to zone 21 at the location -123.43,1200.03,21.34.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::rebind(21,-123.43,1200.03,21.34);
    }
}
```

quest::flagcheck

Explanation: Checks the value of a flag, usually used for quests.

Full Command: quest::flagcheck(flag,value)

Flag = The number of the flag that you are checking.

Value = The value you are looking for when checking the flag.

Example:

After hailing the NPC will check if flag 1 is set to 0.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::flagcheck(1,0);
    }
}
```

quest::write

Explanation: Writes some text to a file on the hard drive

Full Command: `quest::write(file,"text")`

File = The name of the file you want to write the text to.

Text = The text you want to write to the file chosen.

Example:

After hailing the NPC a file names hi.txt is written with the contents "Hail" in it.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::write("hi.txt","Hail");
    }
}
```

quest::settime

Explanation: Changes the time in the zone changing the sky and lighting depending on the time set.

Full Command: quest::settime(hour,minutes)

Hour = The hour of the day in a 24 hour schedule (1:00pm = 13:00)

Minutes = The Minutes into the hour set before it.

Example:

After hailing the NPC the time is set to five after midnight.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::settime(0,5);
    }
}
```

quest::setsky

Explanation: Changes the color of the sky.

Full Command: quest::setsky(n)

n = Anywhere from 0 to 255.

Example:

After hailing the NPC the sky goes deep red.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::setsky(250);
    }
}
```

quest::settimer

Explanation: Starts a timer that counts seconds.

Full Command: `quest::(timerid,seconds)`

Timerid = The ID number of the timer you would like to start. Found in the database.

Seconds = The amount of time you want the timer to run for in seconds.

Example:

After hailing the NPC timer 1 is set to run for 3 minutes (180 seconds)

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::settimer(1,180);
    }
}
```

quest::stoptimer

Explanation: Stops the chosen timer.

Full Command: quest::stoptimer(timerid)

Timerid = The ID of the timer you would like to stop. Can be found in your database.

Example:

After hailing the NPC the time with the ID 1 is stopped.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::stoptimer(1);
    }
}
```

quest::settarget

Explanation: Sets a mobs target.

Full Command: `quest::settarget(type,ID)`

Type =

ID = The ID of the NPC or player to target.

Example:

quest::follow

Explanation: Makes the mob or NPC follow a chosen NPC or player.

Full Command: `quest::follow(ID)`

ID = The ID of the NPC or Player that the mob or NPC is supposed to follow.

Example:

After hailing the NPC he follow the NPC with the ID of 1024.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::follow(1024);
    }
}
```

quest::sfollow

Explanation: Stop the mob from following a player or NPC after making it do so with the follow command.

Full Command: quest::sfollow()

Example:

After hailing the NPC again he stops following the player.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::sfollow
    }
}
```

quest::movepc

Explanation: Moves the player to a chosen zone to a chosen location.

Full Command: `quest::movepc(zoneid,x,y,z)`

Zoneid = The ID of the zone you want to move the player to. Can be found in your database.

X = The x Axis you would like to send the player to. Can be found by /loc but switch the x and y.

Y = The y Axis you would like to send the player to. Can be found by /loc but switch the x and y.

Z = The z Axis you would like to send the player to. Can be found by /loc.

Example:

After hailing the NPC the player is moved to zone 23 at location -123.45,1200.03,-34.11.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::movepc(23, -123.45,1200.03,-34.11);
    }
}
```

quest::gmmove

Explanation: Moves the player to a chosen zone to a chosen location.

Full Command: quest::gmmove(zoneid,x,y,z)

Zoneid = The ID of the zone you want to move the player to. Can be found in your database.

X = The x Axis you would like to send the player to. Can be found by /loc but switch the x and y.

Y = The y Axis you would like to send the player to. Can be found by /loc but switch the x and y.

Z = The z Axis you would like to send the player to. Can be found by /loc.

Example:

After hailing the NPC the player is moved to zone 18 at location -123.45,1200.03,-34.11.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::movepc(18, -123.45,1200.03,-34.11);
    }
}
```

quest::movegrp

Explanation: Moves the player and his party to a chosen zone and a chosen location.

Full Command: `quest::movegrp(zoneid,x,y,z)`

Zoneid = The ID of the zone you want to move the party to. Can be found in your database.

X = The x Axis you would like to send the party to. Can be found by /loc but switch the x and y.

Y = The y Axis you would like to send the party to. Can be found by /loc but switch the x and y.

Z = The z Axis you would like to send the party to. Can be found by /loc.

Example:

After hailing the NPC the player and his party is moved to zone 7 at location -123.45,1200.03,-34.11.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::movegrp(7, -123.45,1200.03,-34.11);
    }
}
```

quest::setallskill

Explanation: Sets all the players skills to a certain value.

Full Command: quest::setallskill(n)

n = any number from 0 to 252

Example:

After hailing the NPC all the player skills are set to the maximum of 252..

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::setallskill(252);
    }
}
```

quest::attack

Explanation: Makes the mob or NPC attack anything with the chosen name.

Full Command: quest::attack("name")

Name = The name of the NPC or player to be attacked.

Example:

After hailing the NPC it attacks the player

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::attack($name);
    }
}
```

quest::save

Explanation: Saves the players data to the database.

Full Command: quest::save()

Example:

After hailing the NPC the players data is saved.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::save();
    }
}
```

quest::setglobal

Explanation: Allows you to create or modify an existing global variable for the mob you are using the command on.

Full Command: `quest::setglobal(varname,value,options,duration)`

Varname = The name of the global variable. It's a text name for the variable.

Value = The value of the variable set before it.

Options =

Duration = the length this global variable is to last.

Example:

After hailing the NPC the global variable bigquest is set to 3 for 1 year gametime..

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::setglobal("bigquest","3","0","Y1");
    }
}
```

quest::targlobal

Explanation: This is a quick way to attack a bunch of information in a global variable to a mob.

Full Command: quest::targlobal(varname,value,duration,npclid,charid,zoneid)

Varname = The name of the global variable. It's a text name for the variable.

Value = The value of the variable set before it.

Duration = the length this global variable is to last.

Npclid = The NPC id of the mob the quest is scripted to.

Charid = The ID of the player who has triggered the quest.

Zoneid = The zone this global variable was set in.

Example:

After hailing the NPC the global variable superquest is set to 7 for 3 days gametime.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::targlobal("bigquest","3","D3",$mobid,$user,$zonein);
    }
}
```

quest::delglobal

Explanation: Deletes a global variable.

Full Command: quest::delglobal(varname)

Varname = The name of the variable to be deleted.

Example:

After hailing the NPC the global variable megaquest is deleted.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::delglobal("megaquest");
    }
}
```

quest::spawn2

Explanation: Used to spawn premade mobs or NPCs at a location heading a certain direction and following a certain path.

Full Command: `quest::spawn2(npclid,grid,unused,x,y,z,heading)`

Npclid = The ID of the NPC you want to add to the Spawn2 in your database.

Grid = The grid number you want the NPC to follow.

Unused = This should be left as a "0" as it is unused. Not sure what it does either.

X = x location where the NPC will spawn. Can be found by /loc but switch the x and y.

Y = y location where the NPC will spawn. Can be found by /loc but switch the x and y.

Z = z location where the NPC will spawn. Can be found by /loc but switch the x and y.

Example:

After hailing the NPC an NPC # 1709 is set to spawn and follow grid 13.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::spawn2(1709,13,0,-120.36,1204.02,-48.12,90);
    }
}
```

quest::setstat

Explanation: Sets a chosen character stat to any valid value.

Full Command: quest::setstat(*statid*,*value*)

Statid = See Appendix III (Stats) for the stat you wish to change.

Value = What you wish the chosen stat to be changed to. Max at 305.

Example:

After hailing the NPC the characters base Strength is changed to 200.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::setstat(0,200);
    }
}
```

quest::addloot

Explanation: Specially adds an item to the mobs loot.

Full Command: `quest::addloot(itemid,amountcharges)`

Itemid = The ID of the item you want to add to the mobs loot.

Amountcharge = Amount of the item to give the mob or the charges on the item if applicable.

Example:

After hailing the NPC he gains 1 of item ID 5420 to his inventory.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::addloot(5420,1);
    }
}
```

quest::zone

Explanation: Sends the client a packet of data regarding the zone.

Full Command: quest::zone(**Shortname**)

Shortname = See Appendix IV (Zones) for the zones short name.

Example:

After hailing the NPC the server sends a packet to the client regarding West Freeport.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::zone(freportw);
    }
}
```

quest::shout2

Explanation: Shouts the text in red text no matter what the client has shout text color set as.

Full Command: `quest::shout2("text")`

Text = The text you want to be shouted by the mob or NPC.

Example:

After hailing the NPC shouts in red text.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::shout2("This text is red!");
    }
}
```

quest::changedeity

Explanation: Changes the deity (God) of the character.

Full Command: quest::changedeity(deityid)

Deityid = See Appendix V (Deitys) for the deity ID you wish the character to now worship.

Example:

After hailing the NPC the player converts to Tunarism.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::changedeity(215);
    }
}
```

quest::traindisc

Explanation: Trains the character handing in the tome so that they can use the discipline and it shows up in their disciplines window.

Full Command: quest::traindisc(*tombid*)

Tombid = The Item ID of the discipline tome handed in for training. Can be found in your database.

Example:

After handing the tome it is checked for validity and used to train the character in the correct discipline.

```
sub EVENT_ITEM
{
    If($itemcount(18204) == 1)
    {
        If(quest::isdisc tome($item1))
        {
            # Trains in the correct discipline.
            quest::traindisc($item1);
        }
    }
}
```

quest::isdisc tome

Explanation: Checks to see if an item that was turned in was actually a discipline tome.

Full Command: quest::isdisc tome(**tombid**)

Tombid = The Item ID of the discipline tome handed in for training. Can be found in your database.

Example:

After handing the tome it is checked for validity and used to train the character in the correct discipline.

```
sub EVENT_ITEM
{
    If($itemcount(18204) == 1)
    {
        # Checks if the item id actually a discipline tome.
        If(quest::isdisc tome($item1))
        {
            quest::traindisc($item1);
        }
    }
}
```

quest::surname

Explanation: Changes the last name of the character.

Full Command: quest::surname("newname")

Newname = The last name you want the character to have.

Example:

After hailing the NPC the characters last name is changes to lipshidtz.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::surname("Lipshidtz");
    }
}
```

quest::permaclass

Explanation: Permanently changes the class of the character.

Full Command: quest::permaclass(classid)

Classid = See Appendix XI (Classes) for the class you want the character to permanently become.

Example:

After hailing the NPC the characters becomes a Wizard.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::permaclass(12);
    }
}
```

quest::permarace

Explanation: Permanently changes the Race of the character.

Full Command: quest::permarace(**raceid**)

Raceid = See Appendix VII (Races) for the race you want the character to permanently become.

Example:

After hailing the NPC the characters becomes an Ogre.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::permarace(10);
    }
}
```

quest::permagender

Explanation: Permanently changes the Race of the character.

Full Command: quest::permagender(*genderid*)

Genderid = See Appendix VIII (Gender) for the ID of the gender you want the character to be.

Example:

After hailing the NPC the characters becomes a Neuter.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::permagender(2);
    }
}
```

quest::scribespells

Explanation: Scribes all your class spells available to the character at its current level.

Full Command: quest::scribespells()

Example:

After hailing the NPC the characters scribes all his class spells up to his current level.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::scribespells();
    }
}
```

quest::setlanguage

Explanation: Set the characters ability in the chosen language.

Full Command: `quest::setlanguage(langid,value)`

Langid = See Appendix IX (Languages) for the language ID you want to set on the character.

Value = The value you wish the character language skill to be set to.

Example:

After hailing the NPC the characters becomes fluent in dragon language

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::setlanguage(21,200);
    }
}
```

quest::setskill

Explanation: Sets a chosen skill on the character to a chosen value.

Full Command: quest::setskill(skillid,value);

Skillid = See Appendix II (Skills) for the ID of the skill you wish to change on the character.

Value = The value you wish the chosen skill to be set to.

Example:

After hailing the NPC the characters Swimming is set to 200.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::setskill(50,200);
    }
}
```

quest::itemlink

Explanation: Makes a link in the characters chat box about a chosen item.

Full Command: `quest::itemlink(itemid)`

Itemid = The ID of the item you wish the character to receive a link about. Can be taken from your database.

Example:

After hailing the NPC the characters receives a link of information about item 5054.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::itemlink(5054);
    }
}
```

quest::ding

Explanation: Plays the “DING” sound for the character.

Full Command: `quest::ding()`

Example:

After hailing the NPC the characters receives a link of information about item 5054.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::itemlink(5054);
    }
}
```

quest::start

Explanation: Starts an NPC or mobs walk to a chosen waypoint.

Full Command: `quest::start()`

Example:

After hailing the NPC will walk to the waypoint with the ID of 234.

```
sub EVENT_SAY
{
    if($text =~ /hail/i)
    {
        quest::start(234);
    }
}
```

quest::stop

Explanation: Stops the NPC or mob from walking to waypoints.

Full Command: quest::stop()

Example:

After hailing the NPC will walk to the waypoint with the ID of 234.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::stop();
    }
}
```

quest::pause

Explanation: Pauses the NPC or mob walking to waypoints for a certain amount of seconds.

Full Command: quest::pause(seconds)

Seconds = The amount of time in seconds that you want the NPC or mob to pause for.

Example:

After hailing the NPC will pause for 10 seconds before continuing to the next waypoint.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::pause(10);
    }
}
```

quest::moveto

Explanation: Makes the NPC or mob walk to a certain location.

Full Command: quest::moveto(x,y,z)

X = x location where the NPC will walk to. Can be found by /loc but switch the x and y.

Y = y location where the NPC will walk to. Can be found by /loc but switch the x and y.

Z = z location where the NPC will walk to. Can be found by /loc but switch the x and y.

Example:

After hailing the NPC will walk to the assigned location.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::moveto(120.34,-1034.56,8.41);
    }
}
```

quest::resume

Explanation: Makes the NPC or mob start walking to waypoints again after quest::stop() has stopped them.

Full Command: quest::resume()

Example:

After hailing the NPC will continue its walk to the next waypoint

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::resume();
    }
}
```

quest::addldonpoints

Explanation: Gives the character a Lost dungeons of norrath point for purchasing things from venders who are set to be class 61 (An Adventure Merchant).

Full Command: quest::addldonpoints(**points**,**theme**)

Points = See Appendix X (LDON Themes) for the amount of ldon points you want to give the character.

Theme = The Theme of the Lost dungeon. See list below.

Example:

After hailing the NPC, the character will receive 10 ldon points from the theme Deepest Guk.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::addldonpoints(10,1);
    }
}
```

quest::setnexthevent

Explanation: Sets up a threshold for EVENT_HP to use. When the mob or NPCs Hit Points reach a certain percentage the event specified by EVENT_HP will happen. This is stored as \$hevent.

Full Command: quest::setnexthevent(**percent**)

Percent = The percentage of hit points the NPC or mob will be at to triggers the event to happen.

Example:

After hailing the NPC, it will check it's hit points. If they are below 50% or become lower than 50% he will shout for help.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::setnexthevent(50);
    }
}
Sub EVENT_HP
{
    if($hevent <= 50)
    {
        Quest::shout("Help Me, I'm getting my butt kicked.");
    }
}
```

quest::respawn

Explanation: Respawns a chosen NPC onto a chosen grid.

Full Command: `quest::respawn(npaid,grid);`

Npaid = The ID of the NPC you wish to respawn.

Grid = The grid you wish the NPC to be placed on.

Example:

After hailing the NPC, it will respawn NPC with the id 234 on grid 7.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::respawn(234,7);
    }
}
```

quest::signal

Explanation: Sets a signal to be sent to all the NPCs in the zone with the chosen npcid. After a chosen amount of time. The signal set off the EVENT_SIGNAL events to happen. This command uses Signalid 0.

Full Command: quest::signal(npcid,wait)

Npcid = The ID of the NPC or more if they have the same ID that will fire the EVENT_SIGNAL events.

Wait = An amount in seconds that you want to wait before the signal is sent.

Example:

After hailing the NPC, it will send NPCs with the ID 768 a signal with signalid 0 to fire in 10 seconds. After the signal has fired the NPC will shout "Marco...". You can set up other NPCs the same way to shout back "Polo".

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::signal(768,10);
    }
}
```

This code would be on the NPC with the ID 768

```
Sub EVENT_SIGNAL
{
    Quest::shout("Marco...");
}
```

quest::signalwith

Explanation: Sets a signal to be sent to all the NPCs in the zone with the chosen npcid. After a chosen amount of time. With a certain signaled. That will set off the EVENT_SIGNAL events if the signalid matched the one in the event. The signalid should start at one so it doesn't mess with the quest::signal command. The signalid is assigned to \$signal.

Full Command: quest::signalwith(npcid,signalid,wait)

Npcid = The ID of the NPC or more if they have the same ID that will fire the EVENT_SIGNAL events.

Signalid = The ID of the signal to be fires in EVENT_SIGNAL. Stored in \$signal.

Wait = An amount in seconds that you want to wait before the signal is sent.

Example:

After hailing the NPC, it will send NPCs with the ID 102 a signal with signalid 8 to fire in 30 seconds. After the signal has fired the NPC will shout "...Polo".

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::signalwith(103,8,30);
    }
}
```

This code would be on the NPC with the ID 102

```
Sub EVENT_SIGNAL
{
    If($signal == 8)
    {
        Quest::shout("...Polo");
    }
}
```

quest::set_proximity

Explanation: Set up a proximity or rather a donut shaped bubble around the NPC or mob. That can be checked for players entering and exiting it, allowing a different event on each. EVENT_ENTER and EVENT_EXIT use this to see if they should fire their events. You may only set one proximity for each mob or NPC.

Full Command: quest::set_proximity(minX,maxX,minY,maxY,minZ,maxZ)

minX = The Minimum X Axis for the bubble to start at around the NPC or mob.
maxX = The Maximum X Axis for the bubble to start at around the NPC or mob.
minY = The Minimum Y Axis for the bubble to start at around the NPC or mob.
maxY = The Maximum X Axis for the bubble to start at around the NPC or mob.
minZ = The Minimum Z Axis for the bubble to start at around the NPC or mob.
maxZ = The Maximum X Axis for the bubble to start at around the NPC or mob.

Example:

After hailing the NPC, it sets up a proximity around itself so when the character exits it he will ask the character to come back.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::set_proximity(0,20,0,20,0,20);
    }
}

Sub EVENT_EXIT
{
    Quest::shout("Please come back, I'm lonely.");
}
}
```

quest::clear_proximity

Explanation: Clear a set proximity on a mob or NPC.

Full Command: quest::clear_proximity()

Example:

After hailing the NPC, it the proximity that was assigned to it.

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::clear_proximity();
    }
}
```

Events

EVENT_SAY

Explanation: Triggered when the mob or NPC is targeted and the player types something.

Variables Used:

\$text – Any text said to the NPC or mob.

Notes:

(\$text =~/Hello/i) – The /i make the text non case sensitive. So Caps or no-caps it sees it the same. It is always best to include this, that way player don't have to worry about capitalization.

(\$text =~/Hello/) – This would match the player saying "Hello", but not "hello"

(\$text =~/hello/) – This would match the player saying "hello", but not "Hello".

(\$text =~/hello/i) – This would match the player saying "Hello" and "hello".

(\$text =~/me/i) – This would match the "me" in name.

(\$text =~/^bme\b/i) – This would not match the "me" in "name" because the \b means there must not be any text next to the "me" to match it. So it would match " me " but not "name".

(\$text =~/^me\$/i) – This would only match if "me" was the only text said. The ^ means it must be the first thing said and the \$ means it must be the last thing said. So it would match "me" but not "kill me" or "me big man".

Example:

The Mob or NPC will react with the following text when the player hails him or if the player says tent.

```
sub EVENT_SAY
{
    if($text =~/Hail/i)
    {
        quest::say("I wish I had a [tent] to sell me good under.");
    }
    if($text =~/tent/i)
    {
        quest::say("Yeah, I hear it is to rain later in the day.");
    }
}
```

EVENT_ITEM

Explanation: Triggered when the player gives the mob or NPC something.

Variables Used:

- \$item1** - The itemid of the item handed to the NPC or mob in the first slot.
- \$item2** - The itemid of the item handed to the NPC or mob in the second slot.
- \$item3** - The itemid of the item handed to the NPC or mob in the third slot.
- \$item4** - The itemid of the item handed to the NPC or mob in the fourth slot.

Notes:

The quest system will only see items if they are handed to the NPC or mob separate in the four available slots. If you were to give the NPC or mob 4 bone chips in a stack in the first slot, it would only be counted as one. You must split stacks up into the 4 separate slots.

Example:

```
# When the player gives the NPC or mob 2 item with the ID of 13916 the  
# player is rewarded with 8 gold. If the player is not a warrior the  
# NPC or mob returns the items to them and tells them why.
```

```
sub EVENT_ITEM  
{  
    if($itemcount{13916} == 2)  
    {  
        if($class ne 'Warrior')  
        {  
            quest::say("I can only reward Warriors!");  
            quest::summonitem($item1) if($item1);  
            quest::summonitem($item2) if($item2);  
            quest::summonitem($item3) if($item3);  
            quest::summonitem($item4) if($item4);  
        }  
        else  
        {  
            quest::givecash("0","0","8","0");  
        }  
    }  
}
```

EVENT_DEATH

Explanation: Triggered when the NPC or mob dies.

Variables Used: None

Notes: None

Example:

The Mob or NPC shout this line of text as it dies.

```
sub EVENT_DEATH
{
    quest::shout("My brothers shall avenge me, scoundrel!");
}
```

EVENT_ATTACK

Explanation: Triggered when the NPC or mob is attacked.

Variables Used: None

Notes: None

Example:

The Mob or NPC shout this line of text as it attacked.

```
sub EVENT_ATTACK
{
    quest::shout("I will kill you fast, for the gods shall punish you slowly!");
}
```

EVENT_SPAWN

Explanation: Triggered when the NPC or mob is spawned.

Used Variables: None

Notes: None

Example:

The Mob or NPC will add one of the item with the ID 5450 to its loot when it is spawned.

```
sub EVENT_SPAWN
{
    quest::addloot(5450,1);
}
```

EVENT_TIMER

Explanation: Triggered when a timer set by `quest::settimer()` has completed it's time cycle.

Variables Used:

\$timer – This is the TimerID of the timer you wish to set.

\$timername – Used to run the right code based on which timer has completed it cycle, so you can use multiple timers in the same quest.

Notes:

The timer repeats itself forever at the same time interval or until `quest::stoptimer()` has been used.

Example:

When the NPC or mob is hailed he sets timer 1 for 30 seconds after the 30 seconds he sets timer 2 for 30 # seconds. After each timer has run its course it is stopped so the mob or NPC does not continue saying his # assigned text every 30 seconds constantly.

```
sub EVENT_SAY
{
    if($text =~ /hail/i)
    {
        quest::say("I'm going to say HI!!! in 60 seconds.");
        quest::settimer(1, 30);
    }
}

sub EVENT_TIMER
{
    if($timername == 1)
    {
        quest::say("Im about to say it...");
        quest::settimer(2, 30);
        quest::stoptimer(1);
    }
    If($timername == 2)
    {
        quest::say("HI!!!");
        quest::stoptimer(2);
    }
}
```

EVENT_SLAY

Explanation: Triggered when ever the NPC or Mob kills someone.

Variables Used: None

Notes: None

Example:

If the NPC or mob kills someone he will shout the line of text below.

```
sub EVENT_SLAY
{
    quest::shout("And stay down!! Thats what you get for taking on the great Boomba!");
}
```

EVENT_WAYPOINT

Explanation: Triggered when a mob or NPC reaches a waypoint.

Variable Used: None

Notes: None

Example:

At every waypoint the NPC or mob will stop for 10 seconds and call for his kitty.

```
sub EVENT_WAYPOINT
{
    Quest::pause(10);
    quest::say("Kitty... Where are you Kitty...Kitty?");
}
```

EVENT_SIGNAL

Explanation: Triggered by a signal triggered with the quest::signal() or quest::signalwith().

Variables Used:

\$signal – The signal that was sent, which allows multiple signal events per NPC or mob.

Notes:

With the quest::signal() command the variable \$signal is always equal to 0.

Example:

One mob signals another when it is attacked so that the other mob comes to help.

This script is on the first mob, the mob that does the calling for help. He calls for help 5 seconds into the battle. And then screams for help again using signal 2.

```
sub EVENT_ATTACK
{
    questshout::("I am under attack, help me please !!!");
    quest::signal(1134,5);
    quest::signalwith(1134,2,10);
}
```

This script goes on the mob with the NPCID 1134. The mob that is supposed to help the first mob. He denies help on the first signal and uses animation 15 "falling backwards" as an excuse not to help.

```
sub EVENT_SIGNAL
{
    questshout::("Sorry but I can't fight on an empty stomach.");

    if($signal == 2)
    {
        questshout::("Ahhhh, I've been hit by a magic invisible arrow");
        questdoanim(15);
    }
}
```

EVENT_ENTER

Explanation: Triggered by any player who enters the mob or NPC's proximity, set by `quest::set_proximity()`.

Variables Used: None

Notes:

You can only have one proximity per mob or NPC. You can clear the proximity on the NPC or mob by using the `quest::clear_proximity()` command.

Example:

When a player enters the NPC's or mob's proximity he says the line of text below.

```
sub EVENT_SPAWN
{
    quest::set_proximity(0,20,0,20,0,20);
}

sub EVENT_ENTER
{
    questshout::("Hey, stranger. What are you doing in this part of town?");
}
```

EVENT_EXIT

Explanation: Triggered by any player who leaves the mobs proximity, set by quest::set_proximity().

Variables Used: None

Notes:

You can only have one proximity per mob or NPC. You can clear the proximity on the NPC or mob by using the quest::clear_proximity() command.

Example:

When a player exits the NPCs or mobs proximity he says the line of text below.

```
sub EVENT_SPAWN
{
    quest::set_proximity(0,20,0,20,0,20);
}

sub EVENT_ENTER
{
    questshout::("Get back here! I'm not done with you.");
}
}
```

EVENT_HP

Explanation: Triggered by a mob or NPCs hit points dropping below a threshold set by `quest::setnextthevent()`.

Variables Used:

`$hpevent` – The percentage of hit points that the event should be fired.

Notes: None

Example:

When a player exits the NPCs or mobs proximity he says the line of text below.

```
sub EVENT_SPAWN
```

```
{
```

```
    quest::setnextthevent(50);
```

```
}
```

```
sub EVENT_HP
```

```
{
```

```
    questshout::("You have only taken half my life. I shall take all of your!!!");
```

```
}
```

EVENT_AGGRO

Explanation: Triggered when a mob or NPC aggro's a player.

Variables Used: None

Notes: None

Example:

When the mob or NPC aggro's the player he shouts this line to text.

```
sub EVENT_AGGRO
{
    questshout::("You're the next one to die !!!");
}
```

Identifier Variables

\$name

Explanation: Returns the first name of the character that triggered the event.

Notes: None

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the players name is Jones Clarik he will say. "Why hello there Jones."*

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say(" Why hello there $name.");
    }
}
```

\$race

Explanation: Returns the race of the character that triggered the event.

Notes: None

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the player is an Iksar he will say "Hello there brother Iksar."*

```
sub EVENT_SAY  
{  
    if($text=~/hail/i)  
    {  
        quest::say("Hello there brother $race.");  
    }  
}
```

\$class

Explanation: Returns the class of the character that triggered the event.

Notes: None

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the player is a Warrior he will say "Are you here for training young Warrior?"*

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("Are you here for training, young $class?");
    }
}
```

\$userid

Explanation: Returns the user ID of the player that triggered the event.

Notes: Can be used for making player specific quests.

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the player is the userid 1 he will say "Applying GMSpeed for user 1 now."*

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("Applying GMSpeed for user $userid, now.");
    }
}
```

\$ulevel

Explanation: Returns the level of the character that triggered the event.

Notes: None

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the player is level 10 he will say "You are level 10 aren't you?"*

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("You are level $ulevel aren't you?");
    }
}
```

\$guildid

Explanation: Returns the guildid of the character that triggered the event.

Notes: Can be used for guild only quests.

Example:

When the mob or NPC is hailed it say the line of text below.

If the player is in a guild with the ID 12 he will say "Welcome all guild 12 members"

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("Welcome all guild $guildid members.");
    }
}
```

\$guildrank

Explanation: Returns the guild rank of the character that triggered the event.

Notes: Can be used for guild offer only quests.

Example:

When the mob or NPC is hailed it say the line of text below.

If the player is in a guild rank of 5 he will say "Only guild rank 5 allowed in here."

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("Only guild rank $guildrank allowed in here.");
    }
}
```

\$mname

Explanation: Returns the name of the mob or NPC that the quest is triggered on.

Notes: None

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the mob or NPC who this quest is assigned to is named Morgo
then he will say "Morgo doesn't like humans."*

```
sub EVENT_SAY  
{  
    if($text=~/hail/i)  
    {  
        quest::say("$mname doesn't like humans.");  
    }  
}
```

\$mobid

Explanation: Returns the NPCTypeID of the mob that the player triggered the event on.

Note: This is not the NPCid of the general mob it is the exact ID of the individual mob. There can be more than one mob with the same name but only one mob with this exact NPCTypeID.

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the mob or NPC who this quest is assigned to has the NPCTypeID of 15643
then he will say "Call me later on my personal NPCTypeID, its 15643"*

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("Call me later on my personal NPCTypeID, its $mobid.");
    }
}
```

\$mlevel

Explanation: Returns the level of the mob that the character triggered the event on.

Notes: None

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the mob or NPC who this quest is assigned to is level 20
then he will say "You can't beat me, I'm level 20."*

```
sub EVENT_SAY  
{  
    if($text=~/hail/i)  
    {  
        quest::say("You can't beat me, I'm level $mlevel.");  
    }  
}
```

\$faction

Explanation: Returns the faction level the player has towards the NPC or mob main faction.

Notes:

This is how the faction numbers are to be translated to the faction standing.

- 1 – Ally
- 2 – Kindly
- 3 – Warmly
- 4 – Amiably
- 5 – Indifferent
- 6 – Scowls
- 7 – Threatening
- 8 – Dubious
- 9 - Apprehensive

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the mob or NPC who this quest is assigned to has Amiable Faction
#towards the character then he will say “Amiable faction is a 4 on my scale.”*

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say(“Amiable faction is a $faction on my scale.”);
    }
}
```

\$zoneid

Explanation: Returns the zoneID that the event occurred in.

Notes:

You can find the ZoneID of each zone in your database.

Example:

```
# When the mob or NPC is hailed it say the line of text below.  
# If the mob or NPC is in the zone with the ID 24 then  
# the NPC or mob will say "It feels like zoneid 24 in here today."
```

```
sub EVENT_SAY  
{  
    if($text=~/hail/i)  
    {  
        quest::say("it feels like zoneid $zoneid in here today.");  
    }  
}
```

\$zonesn

Explanation: Returns the zone short name that the event occurred in.

Notes: None

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the mob or NPC is in the zone West Freeport then
the NPC or mob will say "It's a nice day in freportw today isn't it?"*

```
sub EVENT_SAY  
{  
    if($text=~/hail/i)  
    {  
        quest::say("It's a nice day in $zonesn today isn't it?");  
    }  
}
```

\$zoneIn

Explanation: Returns the long name of the zone that the event occurred in.

Notes: None

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the mob or NPC is in the zone West Freeport then
the NPC or mob will say "It's a nice day in West Freeport today isn't it?"*

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("It's a nice day in $zoneIn today isn't it?");
    }
}
```

\$status

Explanation: Returns the account status of the user who triggered the event.

Notes:

Can be used to make Guide or GM only quests. This is based on your over all status on the server, if you are a normal player, guide, or gm.

0 - Normal User
10 - Priviledged User
20 - Very Priviledged User
80 - Guide
100 - GM
150 - Lead GM
200 - Server OP
250 - Debug User

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the user is status level 250 (Server Op) then
the NPC or mob will say "I wish I had 250 status on this server!"*

```
sub EVENT_SAY  
{  
    if($text=~/hail/i)  
    {  
        quest::say("I wish I had $status status on this serer!");  
    }  
}
```

\$item1 , \$item2 , \$item3 , \$item4

Explanation: Returns the ItemIDs of items given by the player to the mob that even was triggered on.

Notes:

You must split stacks of items in single items or they will only be counted as one item being handed in. To hand in 4 of the same item just place one of each into each of the 4 open slots in the give window.

\$item1 - The object in the first give slot of the give window.

\$item2 - The object in the second give slot of the give window.

\$item3 - The object in the third give slot of the give window.

\$item4 - The object in the fourth give slot of the give window.

Example:

When the mob or NPC is hailed it say the line of text below.

If the user hands the NPC or mob an item ID 1034 into slot 1 then

the NPC or mob will say "Thanks for the Item 1034, I was looking for one of those."

```
sub EVENT_ITEM
```

```
{
```

```
    if($item1== 1034)
```

```
    {
```

```
        quest::say("Thanks for the Item $item1, I was looking for one of those.");
```

```
    }
```

```
}
```

\$itemcount{itemid}

Explanation: Counts how many items of the specified ID was given to the Mob or NPC.

Variables Used:

Itemid – The itemID of the item you wish to get the total that was given to the NPC or mob

Notes:

Used to see if the character handed in the correct amount of one item.

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the user hands the NPC or mob 2 items with the ID 1001 then
the NPC or mob will say "Wow, 2 of the same 1001 item, thanks."*

```
sub EVENT_ITEM
{
    if($itemcount{1001} == 2)
    {
        quest::say("Wow, $itemcount of the same 1001 item,thanks.");
    }
}
```

\$copper

Explanation: Returns the number of copper coins given to the mob or NPC.

Notes: None

Exmample:

When the mob or NPC is hailed it say the line of text below.

If the user hands the NPC or mob 5 Copper coins then

the NPC or mob will say "Oh what a generous person, 5 copper."

```
sub EVENT_ITEM
```

```
{
```

```
    if($copper == 5)
```

```
    {
```

```
        quest::say("Oh what a generous person, $copper copper.");
```

```
    }
```

```
}
```

\$silver

Explanation: Returns the number of silver coins given to the mob or NPC.

Notes: None

Exmample:

*# When the mob or NPC is hailed it say the line of text below.
If the user hands the NPC or mob 5 Silver coins then
theNPC or mob will say "Oh what a generous person, 5 silver."*

```
sub EVENT_ITEM  
{  
    if($silver == 5)  
    {  
        quest::say("Oh what a generous person, $silver silver.");  
    }  
}
```

\$gold

Explanation: Returns the number of gold coins given to the mob or NPC.

Notes: None

Exmample:

*# When the mob or NPC is hailed it say the line of text below.
If the user hands the NPC or mob 5 Gold coins then
the NPC or mob will say "Oh what a generous person, 5 gold."*

```
sub EVENT_ITEM  
{  
    if($gold == 5)  
    {  
        quest::say("Oh what a generous person, $gold gold.");  
    }  
}
```

\$platinum

Explanation: Returns the number of copper coins given to the mob or NPC.

Notes: None

Exmample:

*# When the mob or NPC is hailed it say the line of text below.
If the user hands the NPC or mob 5 Platinum coins then
the NPC or mob will say "Oh what a generous person, 5 platinum."*

```
sub EVENT_ITEM  
{  
    if($platinum == 5)  
    {  
        quest::say("Oh what a generous person, $platinum platinum.");  
    }  
}
```

\$x , \$y , \$z

Explanation: Returns the coordinates of the mob or NPC.

Notes:

Each coordinate can be used separately.

\$x – Is the x Coordinates of the mob or NPC.

\$y – Is the y Coordinates of the mob or NPC.

\$z – Is the z Coordinates of the mob or NPC.

Exmample:

When the mob or NPC is hailed it say the line of text below.

If the NPC or mob is at location 124,-1345,28 then

the NPC or mob will say "I'm standing at location 124,-1345,28."

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("I'm standing at location $x,$y,$z.");
    }
}
```

\$h

Explanation: Returns the heading of the mob or NPC.

Notes:

Headings can range from 0 to 260. This is a general breakdown of the values.

260 - 20 = "North"
21 - 40 = "North West"
41 - 80 = "West"
81 - 100 = "South West"
101 - 150 = "South"
151 - 170 = "South East"
171 - 210 = "East"
211 - 240 = "North East"

Example:

*# When the mob or NPC is hailed it say the line of text below.
If the NPC or mob is facing 47(west) then
the NPC or mob will say "I'm facing 47, which is west."*

```
sub EVENT_SAY
{
    if($text=~/hail/i)
    {
        quest::say("I'm facing $h, which is west.");
    }
}
```

\$text

Explanation: Returns all speaking responses from the character to the mob or NPC.

Notes:

(\$text =~/Hello/i) – The /i make the text non case sensitive. So Caps or no-caps it sees it the same. It is always best to include this, that way player don't have to worry about capitalization.

(\$text =~/Hello/) – This would match the player saying "Hello", but not "hello"

(\$text =~/hello/) – This would match the player saying "hello", but not "Hello".

(\$text =~/hello/i) – This would match the player saying "Hello" and "hello".

(\$text =~/me/i) – This would match the "me" in name.

(\$text =~/bme\b/i) – This would not match the "me" in "name" because the \b means there must not be any text next to the "me" to match it. So it would match " me " but not "name".

(\$text =~/^me\$/i) – This would only match if "me" was the only text said. The ^ means it must be the first thing said and the \$ means it must be the last thing said. So it would match "me" but not "kill me" or "me big man".

Example:

*# If the character says "the owl flies at midnight" then
the NPC or mob will say "Going my way?."*

```
sub EVENT_SAY
{
    if($text =~/the owl flies at midnight/i)
    {
        quest::say("Going my way?");
    }
}
```

Operators

== : Same or equal to

Explanation: If variable \$1 is the same as variable \$2

!= : Not the same or not equal

Explanation: If variable \$1 is not the same as variable \$2

> : Greater than

Explanation: If variable \$1 is more than equal to variable \$2

< : Less Than

Explanation: If variable \$1 is less than equal to variable \$2

>= : Greater than or equal to

Explanation: If variable \$1 is more than or equal to variable 2\$

<= : Less than or equal to

Explanation: If variable \$1 is less than or equal to variable \$2

&& : And

Explanation: If variable \$1 and Variable \$2 are (any of the operators above) then continue.

|| : Or

Explanation: If variable \$1 or Variable \$2 are (any of the operators above) then continue.

Appendix I (Animations)

- 1 - Kick
- 2 - Piercing
- 3 - 2h slash
- 4 - 2h blunt/piercing
- 5 - Throw
- 6 - Offhand attack
- 7 - Bash/Dragon Punch
- 8 - Mainhand slash
- 9 - Shoot bow
- 10 - Swim
- 11 - Round Kick
- 12 - Shake head no
- 13 - Embarrassed(looks down)
- 14 - Embarrassed(looks down)
- 15 - Falling backwards animation
- 16 - Death animation
- 17 - Elbows at waist, hands pointing out
- 18 - Hands pointing straight down
- 19 - Run animation(I think)
- 20 - Jump
- 21 - Falling face first
- 22 - Duck walking
- 23 - No idea
- 24 - Bow
- 25 - Swimming in place
- 26 - Idle animation(troll scratches butt, etc)
- 27 - Cheer
- 28 - Disgusted
- 29 - Wave
- 30 - Rude
- 31 - Yawn
- 32 - No emote(?)
- 33 - Sit down
- 34 - Walk backwards(I think)
- 35 - No emote(?)
- 36 - Kneel
- 37 - Swim
- 38 - Sitting down
- 39 - No emote(?)
- 40 - No emote(?)
- 41 - No emote(?)
- 42 - Cast anim(Canni, etc)
- 43 - Cast anim(Buffs, etc)
- 44 - Cast anim(Lifetap, nuke, etc)

45 - Flying Kick
46 - Tiger Claw
47 - Eagle Strike
48 - Nod yes
49 - Shake head no
50 - Plead
51 - Applaud/clap
52 - No idea
53 - Blush
54 - Chuckle
55 - No idea
56 - Duck(the emote)
57 - Curious look
58 - Dance
59 - Disagree
60 - Glare
61 - Peer
62 - Kneel
63 - Laugh
64 - Point
65 - Shrug
66 - Raise hand
67 - Salute
68 - Shiver
69 - Tap Foot
70 - Bow
71 - None(?)
72 - None(?)
73 - Sit down

Appendix II (Skills)

- 0 - 1H Blunt
- 1 - 1H Slashing
- 2 - 2H Blunt
- 3 - 2H Slashing
- 4 - Abjuration
- 5 - Alteration
- 6 - Apply Poison
- 7 - Archery
- 8 - Backstab
- 9 - Bind Wound
- 10 - Bash
- 11 - Block
- 12 - Brass Instruments
- 13 - Channeling
- 14 - Conjuraton
- 15 - Defense
- 16 - Disarm
- 17 - Disarm Traps
- 18 - Divination
- 19 - Dodge
- 20 - Double Attack
- 21 - Dragon Punch
- 22 - Duel Wield
- 23 - Eagle Strike
- 24 - Evocation
- 25 - Feign Death
- 26 - Flying Kick
- 27 - Forage
- 28 - Hand To Hand
- 29 - Hide
- 30 - Kick
- 31 - Meditate
- 32 - Mend
- 33 - Offense
- 34 - Parry
- 35 - Pick Lock
- 36 - Piercing
- 37 - Riposte
- 38 - Round Kick
- 39 - Safe Fall
- 40 - Sense Heading
- 41 - Sing
- 42 - Sneak
- 43 - Specialize Abjure
- 44 - Specialize Alteration

45 - Specialize Conjuraton
46 - Specialize Divination
47 - Specialize Evocation
48 - Pick Pockets
49 - Stringed Instruments
50 - Swimming
51 - Throwing
52 - Tiger Claw
53 - Tracking
54 - Wind Instruments
55 - Fishing
56 - Make Poison
57 - Tinkering
58 - Research
59 - Alchemy
60 - Baking
61 - Tailoring
62 - Sense Traps
63 - Blacksmithing
64 - Fletching
65 - Brewing
66 - Alcohol Tolerance
67 - Begging
68 - Jewelry Making
69 - Pottery
70 - Percussion Instruments
71 - Intimidation
72 - Berserking
73 - Taunt

Appendix III (Stats)

- 0 - Strength
- 1 - Stamina
- 2 - Agility
- 3 - Dexterity
- 4 - Intelligence
- 5 - Wisdom
- 6 - Charisma

Appendix IV (Zones)

Short Name	Long Name
acrylia	Acrylia Caverns
airplane	Plane of Air
akanon	Ak'Anon
akheva	Akheva Ruins
arena	The Arena
bazaar	The Bazaar
befallen	Befallen
beholder	Gorge of King Xorbb
blackburrow	Blackburrow
bothunder	Bastion of Thunder
burningwood	The Burning Wood
butcher	Butcherblock Mountains
cabeast	Cabilis East
cabwest	Cabilis West
cauldron	Dagnor's Cauldron
cacizthule	Lost Temple of Caciz'Thule
charasis	Howling stones
chardok	Chardok
citymist	The City of Mist
cobaltscar	Cobalt Scar
commons	West Commonlands
crushbone	Crushbone
crystal	Crystal Caverns
cshome	Sunset Home
dalnir	Dalnir
dawnshroud	Dawnshroud Peaks
dreadlands	Dreadlands
droga	Mines of Droga
eastkarana	Eastern Plains of Karana
eastwastes	Eastern Wastelands
echo	Echo Caverns
ecommons	East Commonlands
emeraldjungle	The Emerald Jungle
erudnext	Erudin
erudnint	Erudin Palace
erudxing	Erud's Crossing
everfrost	Everfrost
fearplane	Plane of Fear
ferrott	The Ferrott

felwithea	Northern Felwithe
felwitheb	Southern Felwithe
fieldofbone	Field of Bone
firiona	Firiona Vie
freporte	East Freeport
freportn	North Freeport
freportw	West Freeport
frontiermtns	Frontier Mountains
frozenshadow	Tower of Frozen Shadow
fungusgrove	Fungus Grove
gfaydark	Greater Faydark
greatdivide	Great Divide
griegsend	Grieg's End
grimling	Grimling Forest
grobb	Grobb
growthplane	Plane of Growth
gukbottom	Ruins of Old Guk
guktop	Guk
halas	Halas
hateplane	Plane of Hate
highkeep	High Keep
highpass	Highpass Hold
hohonora	Halls of Honor
hole	The Hole
hollowshade	Hollowshade Moor
iceclad	Iceclad Ocean
innothule	Innothule Swamp
kael	Kael Drakael
kaesora	Kaesora
kaladima	North Kaladim
kaladimb	South Kaladim
karnor	Karnor's Castle
katta	Katta Castellum
kedge	Kedge Keep
kerraridge	Kerra Isle
kithicor	Kithicor Woods
kurn	Kurn's Tower
lakeofillomen	Lake of Ill Omen
lakerathe	Lake Rathetear
lavastorm	Lavastorm Mountains
letalis	Mons Letalis
lfaydark	Lesser Faydark
load	Loading Zone
maiden	The Maiden's Eye
mischieffplane	Plane of Mischief
mistmoore	Castle Mistmoore
misty	Misty Thicket
mseru	Marus Seru
najena	Najena
necropolis	Dragon Necropolis

nektulos	Nektulos Forest
neriaka	Neriak Foreign Quarter
neriakb	Neriak Commons
neriakc	Neriak Third Gate
netherbian	Netherbian Lair
nexus	The Nexus
northkarana	Northern Plains of Karana
nro	Northern Desert of Ro
nurga	Mines of Nurga
oasis	Oasis of Marr
oggok	Oggok
oot	Ocean of Tears
overthere	The Overthere
paineel	Paineel
paludal	Paludal Caverns
paw	Lair of the Splitpaw
permafrost	Permafrost Caverns
powar	Plane of War
poearth	Plane of Earth
pofire	Doomfire, Burning Lands
powater	Plane of Water
poinnovation	Plane of Innovation
podisease	Plane of Disease
pojustice	Plane of Justice
ponightmare	Plane of Nightmare
potranquility	Plane of Tranquility
potimea	Plane of Time A
potimeb	Plane of Time B
povalor	Plane of Valor
postorms	Plane of Storms
poknowledge	Plane of Knowledge
potorment	Plane of Torment
qcat	Qeynos Aqueduct System
qey2hh1	Western Plains of Karana
qeynos	South Qeynos
qeynos2	North Qeynos
qeytoqrg	Qeynos Hills
qrg	Surefall Glade
rathemtn	Rathe Mountains
rivervale	Rivervale
runnyeye	Runnyeye Citadel
scarlet	The Scarlet Desert
sebilis	Old Sebilis
shadeweaver	Shadeweavers Thicket
shadowhaven	Shadow Haven
sharvahl	Shar Vahl
sirens	Sirens Grotto
skyfire	Skyfire Mountains
skyshrine	Skyshrine
sleeper	Sleepers Tomb

soldunga	Solusek's Eye
soldungb	Nagafen's Lair
solrotower	Tower of Solusek Ro
soltemple	Temple of Solusek Ro
southkarana	Southern Plains of Karana
sro	Southern Desert of Ro
sseru	Sanctus Seru
ssratemple	Ssraeshza Temple
steamfont	Steamfont Mountains
stonebrunt	Stonebrunt Mountains
swampofnohope	Swamp Of No Hope
templeveeshan	Temple of Veeshan
tenebrous	The Tenebrous Mountains
thedeep	The Deepshade
thegrey	The Grey
thurgadina	City of Thurgadin
thurgadinb	Icewell Keep
timorous	Timorous Deep
tox	Toxxulia Forest
trakanon	Trakanon's Teeth
tutorial	The Tutorial Zone
twilight	The Twilight Sea
umbral	The Umbral Plains
unrest	Estate of Unrest
veeshan	Veeshan's Peak
velketor	Velketor's Labrynth
vexthal	Vex Thal
wakening	The Wakening Lands
warrens	Warrens
warslikewood	Warslik's Woods
westwastes	Western Wastelands

Appendix V (Deity)

- 201 - Bertoxulous (The Plague Bringer) - Evil
- 202 - Brell Seliris (The Duke of Below) - Good
- 203 - Cazic Thule (The Faceless) - Evil
- 204 - Erollisi Marr (The Queen of Love) - Good
- 205 - Bristlebane (The King of Thieves) - Neutral
- 206 - Innoruuk (The Prince of Hate) - Evil
- 207 - Karana (The Rain Keeper) - Good
- 208 - Mithaniel Marr (The Truthbringer Also known as the Lightbringer) - Good
- 209 - Prexus (The Ocean Lord) - Good
- 210 - Quellious (The Tranquil) - Good
- 211 - Rallos Zek (The WarLord) - Neutral
- 212 - Rodcet Nife (The Prime Healer) - Good
- 213 - Solusek Ro (The Burning Prince) - Neutral
- 214 - The Tribunal (The Council of Justice) - Neutral
- 215 - Tunare (The Mother of All) - Good
- 216 - Veeshan (The Mother of all Wurms also known as the Wurm Queen) - Neutral
- 396 - Agnostic - No Religion

Appendix VI (Classes)

1 Warrior
2 Cleric
3 Paladin
4 Ranger
5 Shadow Knight
6 Druid
7 Monk
8 Bard
9 Rogue
10 Shaman
11 Necromancer
12 Wizard
13 Magician
14 Enchanter
15 Beastlord
16 Berserker
20 GM Warrior
21 GM Cleric
22 GM Paladin
23 GM Ranger
24 GM ShadowKnight
25 GM Druid
26 GM Monk
27 GM Bard
28 GM Rogue
29 GM Shaman
30 GM Necromancer
31 GM Wizard
32 GM Magician
33 GM Enchanter
34 GM Beastlord
35 GM Berserker
40 Banker
41 Shopkeeper
60 Adventure Recruiter
61 Adventure Merchant
63 Tribute Master

Appendix VII (Races)

- 0 Soldier
- 1 Human
- 2 Barbarian
- 3 Erudite
- 4 Wood Elf
- 5 High Elf
- 6 Dark Elf
- 7 Half Elf
- 8 Dwarf
- 9 Troll
- 10 Ogre
- 11 Halfling
- 12 Gnome
- 13 Aviak
- 14 Were Wolf
- 15 Brownie
- 16 Centaur
- 17 Golem
- 18 Giant / Cyclops
- 19 Trakenon
- 20 Doppelganger
- 21 Evil Eye
- 22 Beetle
- 23 Kerra
- 24 Fish
- 25 Fairy
- 26 Froglok
- 27 Froglok Ghoul
- 28 Fungusman
- 29 Gargoyle
- 30 Gasbag
- 31 Gelatinous Cube
- 32 Ghost
- 33 Ghoul
- 34 Giant Bat
- 35 Giant Eel
- 36 Giant Rat
- 37 Giant Snake
- 38 Giant Spider
- 39 Gnoll
- 40 Goblin
- 41 Gorilla
- 42 Wolf
- 43 Bear
- 44 Freeport Guards
- 45 Demi Lich

46 Imp
47 Griffin
48 Kobold
49 Lava Dragon
50 Lion
51 Lizard Man
52 Mimic
53 Minotaur
54 Orc
55 Human Beggar
56 Pixie
57 Dracnid
58 Solusek Ro
59 Bloodgills
60 Skeleton
61 Shark
62 Tunare
63 Tiger
64 Treant
65 Vampire
66 Rallos Zek
67 Highpass Citizen
68 Tentacle
69 Will O Wisp
70 Zombie
71 Qeynos Citizen
72 Ship
73 Launch
74 Piranha
75 Elemental
76 Puma
77 Neriak Citizen
78 Erudite Citizen
79 Bixie
80 Reanimated Hand
81 Rivervale Citizen
82 Scarecrow
83 Skunk
84 Snake Elemental
85 Spectre
86 Sphinx
87 Armadillo
88 Clockwork Gnome
89 Drake
90 Halas Citizen
91 Alligator
92 Grobb Citizen
93 Oggok Citizen
94 Kaladim Citizen
95 Cazic Thule
96 Cockatrice
97 Daisy Man
98 Elf Vampire
99 Denizen
100 Dervish
101 Efreeti

102 Froglok Tadpole
103 Kedge
104 Leech
105 Swordfish
106 Felguard
107 Mammoth
108 Eye of Zomm
109 Wasp
110 Mermaid
111 Harpie
112 Fayguard
113 Drixie
114 Ghost Ship
115 Clam
116 Sea Horse
117 Ghost Dwarf
118 Erudite Ghost
119 Sabertooth Cat
120 Wolf Elemental
121 Gorgon
122 Dragon Skeleton
123 Innoruuk
124 Unicorn
125 Pegasus
126 Djinn
127 Invisible Man
128 Iksar
129 Scorpion
130 Vah Shir
131 Sarnak
132 Draglock
133 Lycanthrope
134 Mosquito
135 Rhino
136 Xalgoz
137 Kunark Goblin
138 Yeti
139 Iksar Citizen
140 Forest Giant
141 Boat
144 Burynai
145 Goo
146 Spectral Sarnak
147 Spectral Iksar
148 Kunark Fish
149 Iksar Scorpion
150 Erollisi
151 Tribunal
152 Bertoxxulous
153 Bristlebane
154 Fay Drake
155 Sarnak Skeleton
156 Ratman
157 Wyvern
158 Wurm
159 Devourer

160 Iksar Golem
161 Iksar Skeleton
162 Man Eating Plant
163 Raptor
164 Sarnak Golem
165 Water Dragon
166 Iksar Hand
167 Succulent
168 Flying Monkey
169 Brontotherium
170 Snow Dervish
171 Dire Wolf
172 Manticore
173 Totem
174 Cold Spectre
175 Enchanted Armor
176 Snow Bunny
177 Walrus
178 Rock-gem Men
181 Yak Man
182 Faun
183 Coldain
184 Velious Dragons
185 Hag
186 Hippogriff
187 Siren
188 Frost Giant
189 Storm Giant
190 Ottermen
191 Walrus Man
192 Clockwork Dragon
193 Abhorent
194 Sea Turtle
195 Black and White Dragons
196 Ghost Dragon
197 Ronnie Test
198 Prismatic Dragon
199 ShikNar
200 Rockhopper
201 Underbulk
202 Grimling
203 Vacuum Worm
204 Evan Test
205 Kahli Shah
206 Owlbear
207 Rhino Beetle,
208 Vampyre
209 Earth Elemental
210 Air Elemental
211 Water Elemental
212 Fire Elemental
213 Wetfang Minnow
214 Thought Horror
215 Tegi
216 Horse
217 Shissar

218 Fungal Fiend
219 Vampire Volatalis
220 StoneGrabber
221 Scarlet Cheetah
222 Zelniak
223 Lightcrawler
224 Shade
225 Sunflower
226 Sun Revenant
227 Shrieker
228 Galorian
229 Netherbian
230 Akheva
231 Spire Spirit
232 Sonic Wolf
233 Ground Shaker
234 Vah Shir Skeleton
235 Mutant Humanoid
236 Seru
237 Recuso
238 Vah Shir King
239 Vah Shir Guard
240 Teleport Man
241 Lujain
242 Naiad
243 Nymph
244 Ent
245 Wrinnfly
246 Tarew Marr
247 Solusek Ro
248 Clockwork Golem
249 Clockwork Brain
250 Spectral Banshee
251 Guard of Justice
252 PoM Castle
253 Disease Boss
254 Solusek Ro Guard
255 Bertoxulous
256 New Tribunal
257 Terris Thule
258 Vegerog
259 Crocodile
260 Bat
261 Slarghilug
262 Tranquilion
263 Tin Soldier
264 Nightmare Wraith
265 Malarian
266 Knight of Pestilence
267 Lepertoloth
268 Bubonian Boss
269 Bubonian Underling
270 Pusling
271 Water Mephit
272 Stormrider
273 Junk Beast

274 Broken Clockwork
275 Giant Clockwork
276 Clockwork Beetle
277 Nightmare Goblin
278 Karana
279 Blood Raven
280 Nightmare Gargoyle
281 Mouths of Insanity
282 Skeletal Horse
283 Saryn
284 Fennin Ro
285 Tormentor
286 Necromancer Priest
287 Nightmare
288 New Rallos Zek
289 Tallon Zek
290 Vallon Zek
291 Air Mephit
292 Earth Mephit
293 Fire Mephit
294 Nightmare Mephit
295 Zebuxoruk
296 Mithaniel Marr
297 Knightmare Rider
298 The Rathe
299 Xegony
300 Balrog
301 ???
302 Lobster Monster
303 Pheonix
304 Tiamat
305 Bear (new)
306 Earth Golem
307 Iron Golem
308 Storm Golem
309 Air Golem
310 Wood Golem
311 Fire Golem
312 Water Golem
313 Veiled Gargoyle
314 Lynx
315 Squid
316 Frog
317 Flying Serpent
318 Tactics Soldier
319 Armored Boar
320 Djinni
321 Boar
322 Knight of Marr
323 Black Armor
324 Nightmare Knight
325 Rallos Ogre
326 Arachnid
327 Crystal Arachnid
328 Tower Model

Appendix VIII (Gender)

0 – Male
1 – Female
2 - Neuter

Appendix IX (Languages)

- 0 - Common Tongue
- 1 - Barbarian
- 2 - Erudian
- 3 - Elvish
- 4 - Dark Elvish
- 5 - Dwarvish
- 6 - Troll
- 7 - Ogre
- 8 - Gnomish
- 9 - Halfling
- 10 - Thieves Cant
- 11 - Old Erudian
- 12 - Elder Elvish
- 13 - Froglok
- 14 - Goblin
- 15 - Gnomish
- 16 - Combine Tongue
- 17 - Elder Teir'Dal
- 18 - Lizardman
- 19 - Orcish
- 20 - Faerie
- 21 - dragon
- 22 - Elder Dragon
- 23 - Dark Speech
- 24 - Vah Shir
- 25 - Unknown1
- 26 - Unknown2

Appendix X (LDON Themes)

- 1= Deepest Guk
- 2= Miragul's Menagerie
- 3= Mistmoore Catacombs
- 4= Rujarkian Hills
- 5= Takish-Hiz